AD-A050 500 NORTHROP CORP HAWTHORNE CALIF ELECTRONICS DIV F/G 17/7
AN/BRN-7 COMPUTER PROGRAM SPECIFICATION. VOLUME XII. COMMON SUB--ETC(U)
OCT 73 N00039-73-C-0209

UNCLASSIFIED NORT-73-48 NL

1 OF 1
AD
A050500

END
DATE
FILMED
3-78
DDC

NORTHROP Electronics Division

☐	This submittal applies to AN/BRN-7 (Submarine Ω) only.
☐	This submittal applies to AN/SRN-( ) (Hydrofoil Ω) only.
☒	This submittal applies to both AN/BRN-7 and AN/SRN-( ).

CONTRACT NO: N00039-73-C-0209

PROGRAM NAME: AN/BRN-7

CDRL No: A01D, A01E, A01F

Title of CDRL: Computer Program Design Specification
Computer Subprogram Design Document
Data Base Design Document

Title of Doc: AN/BRN-7 Computer Program Specification
NORT 73-48
Volume 2 thru 13

Date: 1/16/74

Initial Submittal: ☒ Release Authentication APS 1-16-74

Resubmittal: ☐

ACCESSION for
NTIS White Section ☒
DDC Buff Section ☐
UNANNOUNCED ☐
JUSTIFICATION Per DDC Form 50 on file
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

Prepared by: System Integration Unit
Orgn. Name

A4734 X574
Orgn. No. and Ext.

Approved by:

Norman P. Clark — Supervisor of Preparing Orgn.

Bill Tyler — Program Mgr.

Proj. Engr./Proj. Mgr.

FORM 6512 (2-72)

NORTHROP

Electronics Division

AD-A050500

NORT 73-48

AN/BRN-7 COMPUTER
PROGRAM SPECIFICATION

Volume XII

COMMON SUBROUTINES SUBPROGRAM DESIGN

October 12, 1973

Approved by

G. W. Hauser
Director, Engineering
Navigation Department

Volume XII
of the
AN/BRN-7 OMEGA COMPUTER
PROGRAM SPECIFICATION

CONTENTS

SECTION 1

SCOPE

1.1 IDENTIFICATION

Volume I, Submarine OMEGA Computer Program Performance Specification, defines the functional requirements for the Submarine OMEGA Computer Program which is used by the AN/ARN-99 OMEGA Navigation Set. The Navigation Set and the OMEGA program together comprise the Submarine OMEGA Navigation System. The tape which defines the computer program is entitled AN/BRN-7 Navigation Program.

Volume II, Submarine OMEGA Computer Program Design Specification, allocates the functional requirements of Volume I to the computer routine and sub-program level.

This volume describes the subprogram designated as common subroutines, which has no abbreviation in the listing.

1.2 COMMON SUBROUTINE TASKS

a) Compute Theta 1
Central angle between station and submarine

b) Theta c
Phase prediction, station to submarine

c) D PHI KI
Base to station correction term

d) Position matrix computation
Calculation of R_{ij} matrix

e) R_{ij} rotate
Rotate a 3x3 matrix through two small angles

f) Resolve
Resolve submarine speed to component velocities

g) Bearing
Calculates bearing between system axes and destination

h) Vector by matrix multiply
A 3x3 matrix by a 1x3 vector

i) Cross Product
Vector product

j) DOT Product
Scaler product

k) Square Root
Newton's method applied to square root

l) Sine Cosine

m) Arctangent

n) ArcSine

o) ArcCosine

SECTION 2

APPLICABLE DOCUMENTS

a) Submarine OMEGA Computer Program Performance Specification (Volume I of the Submarine OMEGA Computer Program Specification).

b) Submarine OMEGA Computer Program Design Specification (Volume II of the Submarine OMEGA Computer Program Specification).

c) NORT 68-66, NAP70 User's Manual, July, 1968.

d) NORT 68-115A, Detailed Description of NDC-1070 Computer Instructions, Revision A, February, 1970.

e) NORT 69-87A, NDC-1070 Flow Chart Program, User's Manual.

SECTION 3

REQUIREMENTS

In order to understand the program description contained in the following pages, it is necessary that the reader will have become familiar with the associated functional requirements found in Volume I, Performance Specification, and with the subprogram allocation found in Volume II, Design Specification.

3.1 DETAILED DESCRIPTION

3.1.1 Reference Labels to Flow Diagrams

The code used to reference the particular block in the flow diagrams, Section 3.2, is as follows: The first number, preceded by a p, is the page number found in the upper right corner of the diagrams. This will be followed by a slash sign (/) to separate the page number from the block designator. The designator will either be a mnemonic label (e.g., TEST SYNC), a local label indicated by a dollar sign ($), or an integer. The two types of labels reference the particular information block, on the given page, to which the label is attached. The integer number, n, means that the referenced block is the n^{th} block from the top of the page; p8/3 would refer to page 8 and the third information designation down.

The label p1/$2+3 refers to page 1, and the 3rd information block after the label $ 2. p2/7,8,9 refers to page 2 and the 7th, 8th and 9th blocks.

3.1.2 Flow Diagram Description

a) Compute Theta 1 (Page 1): Computes the earth central angle between a specified station and the submarine. The arguments consist of the station number, with which the station coordinates can be found by table look-up, and a pointer to indicate memory address of the submarine position vector to be used.

   p1/Compute Theta 1:

   The R1 vector is converted to geocentric coordinates by multiplying r_{11} by a^2/b^2

   where a = earth polar radius
   b = earth equatorial radius

p1/2:

The Cross product and DOT product subroutines are used to compute

$$\text{Sin}\theta = \sqrt{(\hat{R}_1 \times \hat{S}) \cdot (\hat{R}_1 \times \hat{S})}$$

$$\text{Cos}\theta = \vec{R1} \bullet \vec{S}$$

p1/3:

$$\theta_1 = \text{TAN}^{-1} \frac{\text{SIN}\theta}{\text{COS}\theta}$$

b) Theta C (Page 2): This routine computes the predicted phase from a station to the submarine for any frequency. The subroutine D PHI KI is used to compensate for the difference in time between reception of signals from the station as opposed to that of the base station. The arguments are θ_1, frequency and station number.

p2/THETA C:

θ_1 is converted to cycles and rescaled.

p2/2:

Obtain θ_2, θ_3 from Propagation Prediction and add to θ_1.

p2/4:

Obtain $\Delta\emptyset$ correction from D PHI KI.

p2/5:

$$\theta_c = \theta_1 + \theta_2 + \theta_3 + \Delta\emptyset.$$

c) D PHI KI (Pages 3, 4): Computes the correction term that represents what is expected to happen to the base station phase between the base station burst time and the time of a non-base station burst for any frequency. Arguments are station number and frequency.

p3/D PHI KI through p4/$3:

The base station reference numbers and the non-base station reference numbers are organized by station and frequency to determine what station is sending on 10.2 kHz when the base station is sending the frequency of the input argument.

$$\begin{pmatrix}\text{Base (or station)}\\ 0-7\end{pmatrix} - \begin{pmatrix}\text{Frequency}\\ 0,1,2\end{pmatrix} = \begin{pmatrix}\text{Reference Number}\\ \text{(Base or Station)}\\ \text{on } 10.2\end{pmatrix} \text{ Modulo } 8$$

p4/$3+1 through p4/$3+5:

This loop decrements the station reference number by unity each time through. The burst collection time is from a table which lists the span of each burst in the pattern. The 0.4 seconds between the End Burst and Start Burst is added. Exit when station reference number is equal to Base Station reference number.

p4/$4:

$$\Delta\phi = \frac{\Delta t}{10}\ \hat{\phi}_{base}$$

d) Position Matrix Computation (Page 5): This routine computes the R_{ij} matrix from a latitude and longitude. Used in the Navigation routine.

e) R_{ij} Rotate (Page 6): Using a small angle, here T2 and T3:

$$\Delta\theta = \begin{bmatrix} \frac{T2^2 + T3^2}{2} & T3 & -T2 \\ -T3 & \frac{-T3^2}{2} & \frac{T2\,T3}{2} \\ T2 & \frac{T2\,T3}{2} & \frac{-T2^2}{2} \end{bmatrix}$$

then

$$[R] = [R] + [\Delta\theta]\ [R]$$

f) Resolve Subroutine (Page 7): The Resolve subroutine is used by velocity processing (Volume VIII) and in the calculation of sea current (Volume X) to compute the uncorrected system velocities V_2 and V_3.

$$V_2 = V_{AT} \operatorname{Sin}(\theta_p + \psi_A) + V_{CT} \operatorname{Cos}(\theta_p + \psi_A)$$

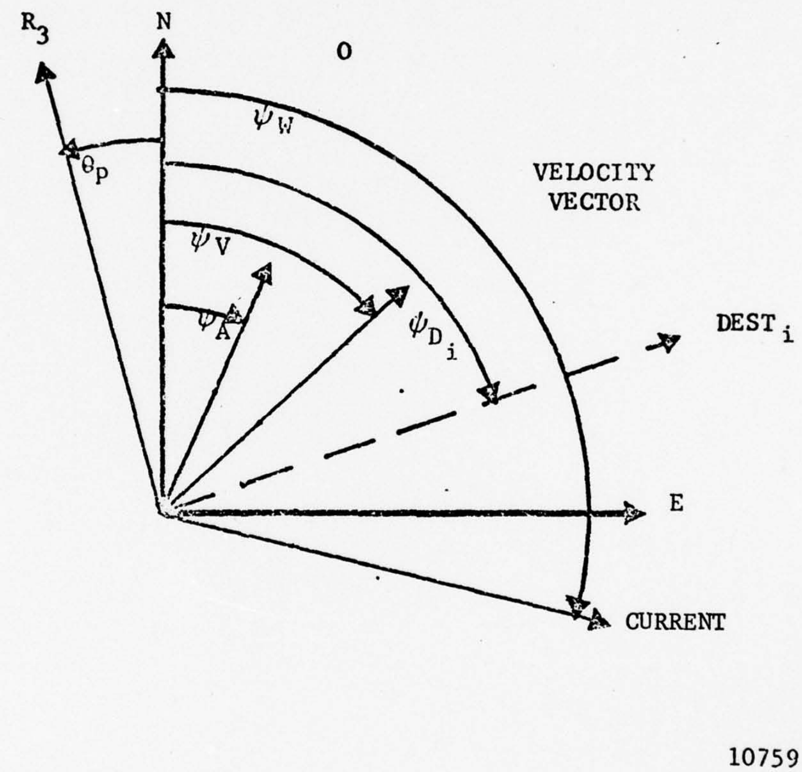
$$V_3 = V_{AT} \operatorname{Cos}(\theta_p + \psi_A) - V_{CT} \operatorname{Sin}(\theta_p + \psi_A)$$

Where $V_{CT} = 0$.

g) Bearing (Page 8): Computes the angle between the R_3 axis and a vector in the plane of R_2, R_3 as indicated below.

$$\theta_p + \psi_{Di} = -\ TAN^{-1}\left(-\frac{\vec{D}_i \bullet \vec{R}_2}{\vec{D}_i \bullet \vec{R}_3}\right)$$

where D is the position vector to Destination i



ψ_A = Submarine Heading HDG

ψ_V = Craft Velocity Track - TK

ψ_D = Course to Selected Destination

ψ_W = Sea Current Direction

θ_P = System Heading N to R_3 CCW

FIGURE 1 SYSTEM AZIMUTHAL RELATIONSHIPS

h) Vector by Matrix Multiply (Page 9): Calculates the multiplication of a 3x3 matrix by a vector V as follows:

$$(V_1, V_2, V_3) \times \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = (X_1, X_2, X_3)$$

$$X_i = \sum_{j=1}^{3} V_j \, r_{ji}$$

i) Cross Product (Page 10): Calculates the vector product of two vectors:

$$\vec{X} = \vec{U} \times \vec{V}$$

$$= (U_2V_3 - U_3V_2)\,\hat{i} + (U_3V_1 - U_1V_3)\,\hat{j} + (U_1V_2 - U_2V_1)\,\hat{k}$$

j) DOT Product (Page 11): Calculates the scaler product of two vectors:

$$\vec{X} = \vec{U} \bullet \vec{V}$$

$$= U_1V_1 + U_2V_2 + U_3V_3$$

k) Square Root (Page 12-13): This routine replaces the long fraction, F, in registers 0 and 1 by its approximate square root. No other register is altered. If F is negative, the square root computation is not carried out, and a negative result is returned. The computation consists of the following steps:

1) F is normalized; i.e., transformed into G by M left shifts, such that $0.5 \leq G < 1$.

2) The first (single-precision) approximation, X_0, is obtained by:

$$X_0 = 0.313567 + (0.890194 - 0.204406 * G) * G$$

3) An improved estimate, X_1, is obtained by a single-precision evaluation of Newton's formula: $X_1 = (X_0 + G/X_0)/2$

4) Newton's formula is evaluated in double-precision to produce the final estimate, X_2.

5) The result is obtained from X_2 by adjusting for the scaling in step 1), thus:

$$F^{1/2} = X_2 * 2^{-M/2}$$

A maximum absolute error less than 2^{-31} is expected, with a negative bias caused by truncation occurring in the right-shift instruction.

The execution time is given by:

296 + 4N microseconds if N is odd (1 through 31).
318 + 4N microseconds if N is even (2 through 30).
where N is the number of leading binary zeros in F.
For F = 0 the time is 54 microseconds.

The storage requirements are 37 words, plus access to an ordered table of powers of two.

1) Sine and Cosine (Pages 14-16): These routines take an angle (scaled π) in registers 0 and 1 and returns results (scaled 2) as follows:

Entry	Result	Other Registers
SIN	sine in 0 and 1	Unchanged
COS	cosine in 0 and 1	Unchanged
SIN-COS	sine in 2 and 3 and cosine in 0 and 1	Reassigned +2

The sine computation takes the following steps:

1) The angle X is rescaled $\pi/2$ by the transformation

$$Y = 2X \text{ if } -1/2 < X < 1/2$$

$$Y = 2\,(1-X) \quad \text{if} \quad 1/2 < X < 1$$

$$Y = 2\,(-1-X) \quad \text{if} \quad -1 < X < -1/2$$

2) If $|Y| < 62/109$, the sine is obtained from the formula:

$$\text{Sin } X = Y * S_3\,(Y^2)$$

(where $S_3(t)$ is a polynomial in t of degree 3).

3) If $|Y| > 62/109$, the sine is obtained by computing

$$|\text{Sin } X| = C_3(Z^2)$$

and adjusting for the sign of X (where $Z = 1 - |Y|$, and $C_3(t)$ is a polynomial in t of degree 3).

The cosine is obtained by computing $\sin \pi(X + 1/2)$.

The maximum relative error is approximately 10^{-8}, or 2^{-27}.

The maximum execution times for the different entries are:

SIN	204 microseconds
COS	208 microseconds
SIN-COS	446 microseconds

The storage requirement is 58 words.

The coefficients for the polynomials S_3 and C_3 were obtained from "Computer Approximations" (Hart et. al., 1967) by making adjustments for scaling to those given under sections SIN 3180 and COS 3700.

$$S_3(t) = 0.785398156180 - 0.322981354578t + 0.039835751436t^2 - 0.002289069985t^3$$

$$C_3(t) = 0.499999995214 - 0.154212373234t + 0.007925896961t^2 - 0.000160303804t^3$$

m) Arctangent (Pages 17, 18): This routine computes the angle A from two (2) arguments S and C of the form K sin A (in registers 2 and 3) and K cos A (in registers 0 and 1). The result, scaled π, is returned in registers 0 and 1, and two (2) registers are pruned.

Use is made of the identify

$$\arctan\left(\frac{\sin t}{\cos t}\right) = \frac{\pi}{4} + \arctan\left(\frac{\sin t - \cos t}{\sin t + \cos t}\right)$$

The computation steps are

1) Transform S and C to a single argument, where

$$X = \frac{|s| - |c|}{|s| + |c|}$$

2) Compute $A = Q/8 + X * T_7 (X^2) * \text{sign}(S) * \text{sign}(C)$
where $T_7(t)$ is a polynomial in t of degree 7, and the "quadrant bits" Q are derived as follows:

Q	S	C
001	+	+
011	+	-
101	-	-
111	-	+

The maximum relative error is less than 10^{-7}, or 2^{-23}.

The maximum execution time is 518 microseconds.

The storage requirement is 51 words.

The coefficients of the polynomial T_7 were obtained from "Computer Approximations" (Hart et. al., 1967) by dividing those given under section ARCTN 4991 by π, and are as follows:

$$T_7(t) = 0.318309854667 - 0.106099021806t + 0.063565605401t^2 - 0.044625393671t^3 + 0.031558174030t^4 - 0.018935106721t^5 + 0.007719779427t^6 - 0.001493916081t^7$$

n) Arcsine (Page 19): This routine computes arcsine by calculating the cosine then using the arctangent subroutine.

o) Arccos (Page 20): Computes the arccosine by calculating the sine then using the arctangent subroutine.

3.2 FLOW DIAGRAMS

The common subroutine flow diagrams are presented on the following pages.

```
*                         COMPUTE THETA1
*
* THIS ROUTINE COMPUTES THE CENTRAL ANGLE BETWEEN ANY STATION AND THE
* CRAFT.  THE ARGUMENTS CONSIST OF THE STATION NUMBER AND A POINTER TO
* THE POSITION VECTOR OF THE CRAFT.
*
```

COMPUTE THETA1

READ R1 VECTOR AND CONVERT TO GEOCENTRIC COORDINATES

↓

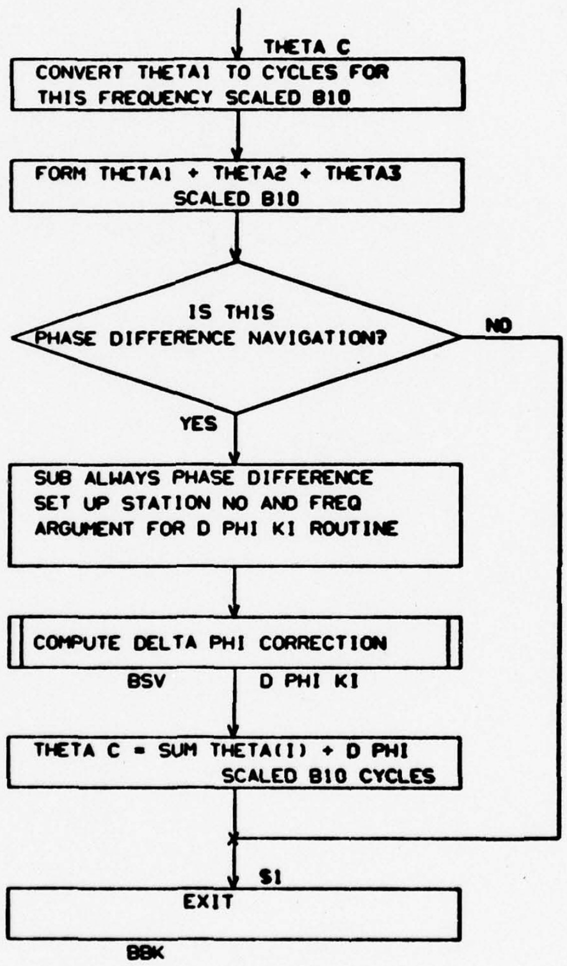
SIN(THETA) = MAG($\hat{R1}$ X $\hat{S}$)   COS(THETA) = R1 . S

↓

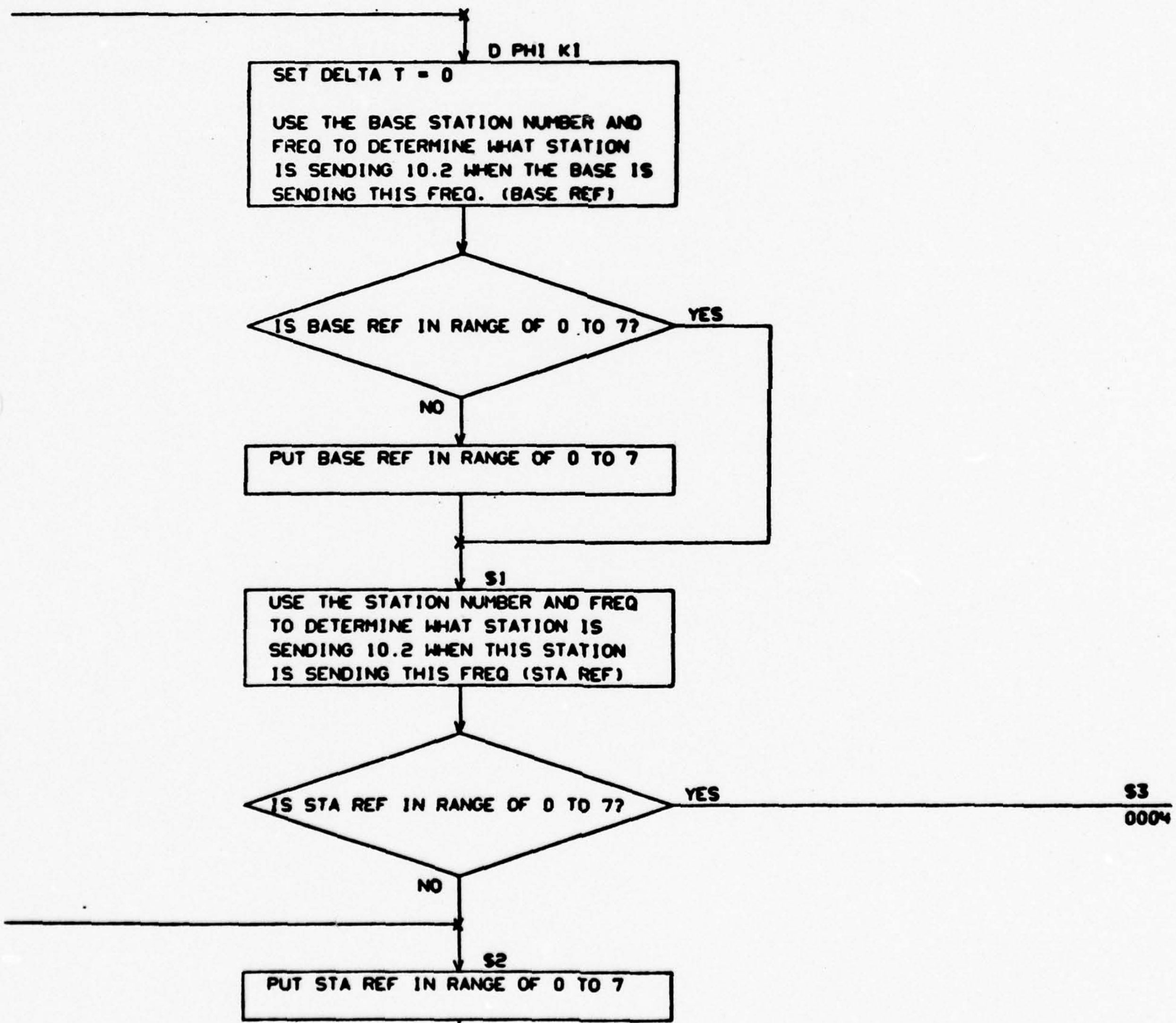
THETA1 = ARCTAN SIN/COS AT PI

BBK

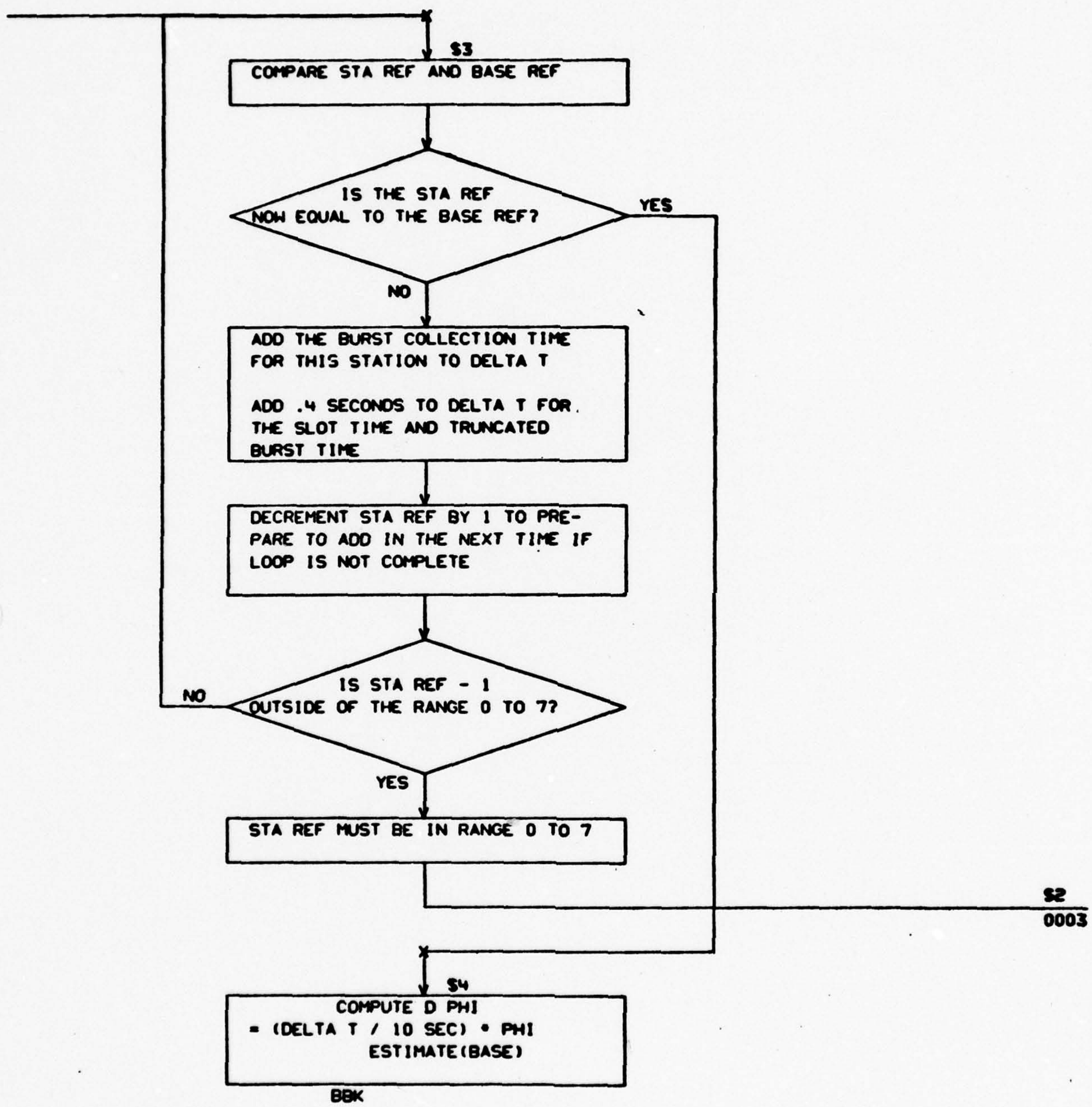
```
*                        THETA C
*
* THIS ROUTINE WILL COMPUTE THE PEDICTED PHASE FROM A STATION TO THE
* CRAFT FOR ANY FREQUENCY.  IN PHASE DIFFERENCE NAVIGATION A CORRECTION
* TERM IS ADDED.  THIS TERM IS THE TIME BETWEEN BASE STATION BURSTS AND
* THIS STATION'S BURSTS TIMES THE PHASE ESTIMATE IN THE BASE STATION
* TRACKING FILTER FOR THIS FREQUENCY.  THE ARGUMENTS CONSIST OF THETA1,
* FREQUENCY AND STATION NUMBER.
*
```



*
D PHI KI
*
* THIS ROUTINE COMPUTES A CORRECTION TERM THAT REPRESENTS WHAT IS
* EXPECTED TO HAPPEN TO THE BASE STATION PHASE BETWEEN THE BASE STATION
* BURST TIME AND THE TIME OF A NON BASE STATION BURST FOR ANY FREQUENCY.
* THIS ROUTINE COMPUTES THE TIME DIFFERENCE AND USES THE PHASE ESTIMATE
* IN THE BASE STATION TRACKING FILTER AS THE EXPECTED CHANGE IN PHASE OF
* THE BASE STATION DURING A TEN SECOND PERIOD. THE ARGUMENTS CONSIST OF
* THE STATION NUMBER AND THE FREQUENCY.
*



* IF THE STATION NUMBER IS THE
* BASE STATION NUMBER THE TEST
* BELOW WILL CAUSE THE CORR-
* ECTION TO BE ZERO


$3
COMPARE STA REF AND BASE REF
IS THE STA REF
NOW EQUAL TO THE BASE REF?
YES
NO
ADD THE BURST COLLECTION TIME
FOR THIS STATION TO DELTA T
ADD .4 SECONDS TO DELTA T FOR
THE SLOT TIME AND TRUNCATED
BURST TIME
DECREMENT STA REF BY 1 TO PRE-
PARE TO ADD IN THE NEXT TIME IF
LOOP IS NOT COMPLETE
IS STA REF - 1
OUTSIDE OF THE RANGE 0 TO 7?
NO
YES
STA REF MUST BE IN RANGE 0 TO 7
$2
0003
$4
COMPUTE D PHI
= (DELTA T / 10 SEC) * PHI
ESTIMATE(BASE)
BBK

```
*                    POSITION MATRIX COMPUTATION
*
* THIS ROUTINES COMPUTES A NINE ELEMENT POSITION MATRIX WITH THE NUMBER
* 3 AXIS POINTING NORTH.  THE ARGUMENTS CONSIST OF THE LATITUDE AND
* LONGITUDE
*
```

```
                  S8
 R11 = SIN (LAT)
 R12 = COS (LAT) * COS (LONG)
 R13 = COS (LAT) * SIN (LONG)
 R21 = 0
 R22 = - SIN (LONG)
 R23 =   COS (LONG)
 R31 =   COS (LAT)
 R32 = - SIN (LAT) * COS (LONG)
 R33 = - SIN (LAT) * SIN (LONG)
         (SCALED B1)
        BBK
```

```
* RIJ ROTATE
*
* THIS PROGRAM WILL ROTATE A 3X3 MATRIX (R)
* THROUGH THE 2 ANGLES SPECIFIED
```

ROTATE RIJS

```
GIVEN THE 2 ANGLES T2 AND T3
THEN
           SQ    T3          -T2
THETA = -T3    -T3*T2/2    T2*T3/2
           T2    T2*T3/2    -T2*T2/2

 WHERE SQ = (T2*T2 + T3*T3) / 2

AND

R(NEW) = R(OLD) + R(OLD)*THETA
```

BSK

```
*
*
* RESOLVE SUBROUTINE
*
* THE RESOLVE SUBROUTINE IS USED BY
* VELOCITY PROCESSING to
* COMPUTE V2 AND V3 FROM SHIPS SPEED WITH
* CROSS TRACK VELOCITY EQUAL TO ZERO.
```

```
                 |
                 v RESOLVE
+----------------------------------------+
| RESOLVE ALONG TRACK VEL (REG 2)        |
| AND CROSS TRACK VEL (REG 3)            |
| THROUGH ANGLE IN REG 0,1+THETA P       |
| TO FORM V2,V3 (RESCALED BY B1)         |
| LEAVE RESULT IN REG 2,3 AFTER          |
| PRUNING 2 REGISTERS                    |
| V2 = VAT*SIN + VCT*COS                 |
| V3 = VAT*COS - VCT*SIN                 |
+----------------------------------------+
          BBK
```

```
*                        BEARING
*
* THIS ROUTINE WILL COMPUTE THE ANGLE BETWEEN THE R3 AXIS AND A VECTOR
* IN THE PLANE OF R2,R3 POINTING AT A FIXED POSITION.  THE ARGUMENTS
* CONSIST OF A POINTER TO THE R2 VECTOR OF THE RIJ MATRIX AND A POINTER
* TO A VECTOR THAT DEFINES THE FIXED POSITION.
*
```

```
                  |
                  v BEARING
 +------------------------------------+
 | COMPUTE BEARING                    |
 |                                    |
 | = - ARCTAN - (D . R2) / (D . R3)   |
 |                                    |
 |                SCALED PI           |
 +------------------------------------+
           BBK
```

```
*                    VECTOR BY MATRIX MULTIPLY
*
* THIS ROUTINE MULTIPLIES A 3X3 MATRIX (R) BY A VECTOR U TO GET A NEW
* VECTOR X.  THE ARGUMENTS CONSIST OF POINTERS TO R, U AND X.
*
```

MATRIX 3D

```
X = V*R

X1 = SUM V(I)*R(I)1, I = 1 TO 3
X2 = SUM V(I)*R(I)2, I = 1 TO 3
X3 = SUM V(I)*R(I)3, I = 1 TO 3
```

BBK

```
*                        CROSS PRODUCT
*
* THIS ROUTINE COMPUTES THE CROSS PRODUCT OF TWO VECTORS U AND V
*
```

REGISTER CROSS

```
X = U X V

X1 = U2*V3 - U3*V2
X2 = U3*V1 - U1*V3
X3 = U1*V2 - U2*V1
```

BBK

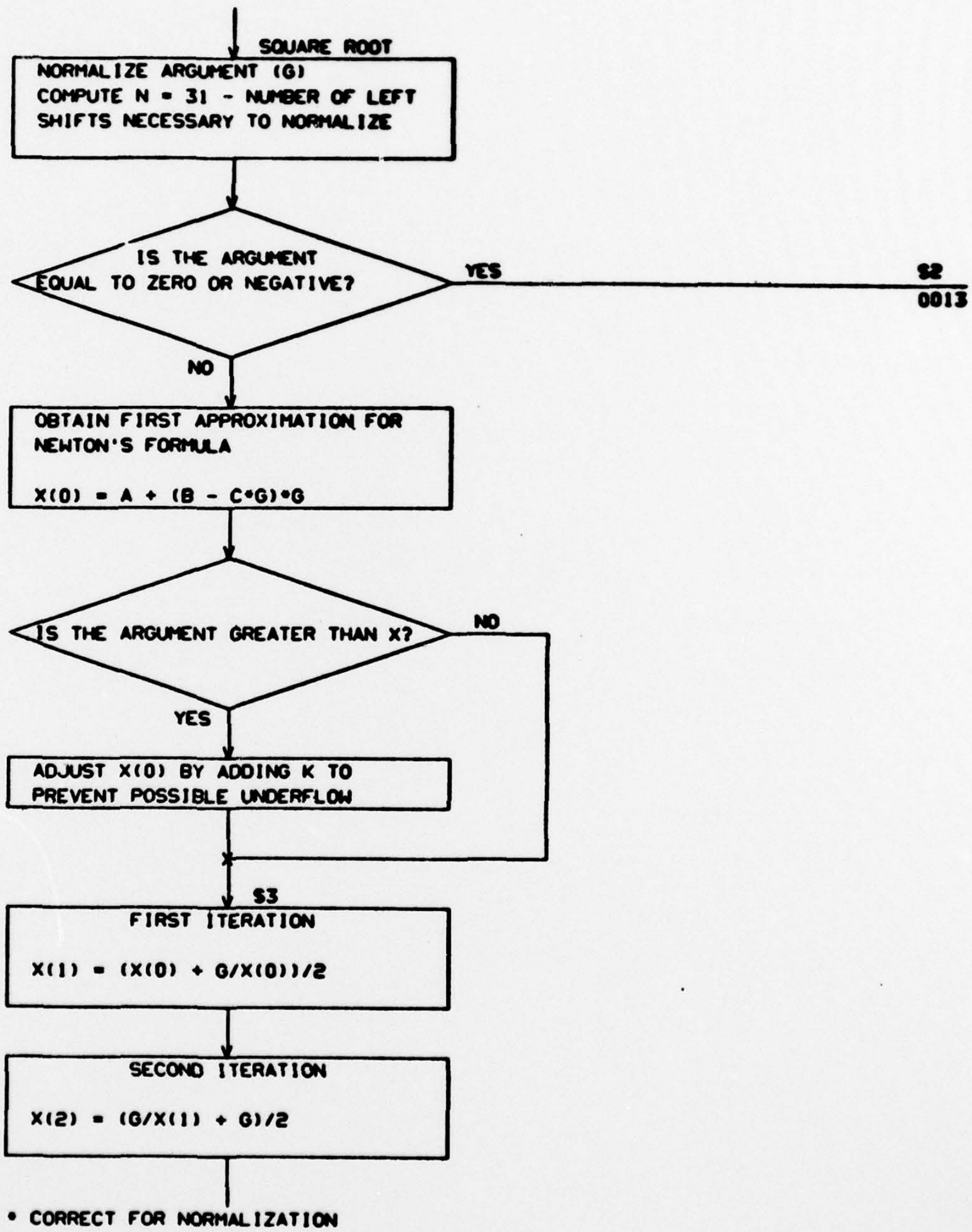
```
*                    DOT PRODUCT
*
* THIS ROUTINE FORMS THE DOT PRODUCT OF THE TWO VECTORS U AND V
*

                 REGISTER DOT
 U . V = SUM U(I)*V(I), I=1 TO 3
        BBK
```

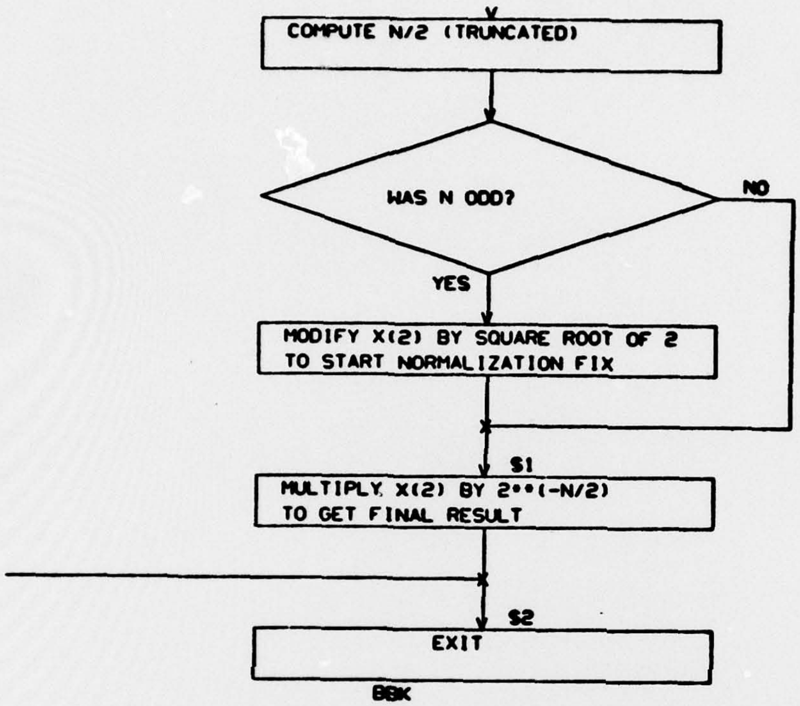
*

* SQUARE ROOT

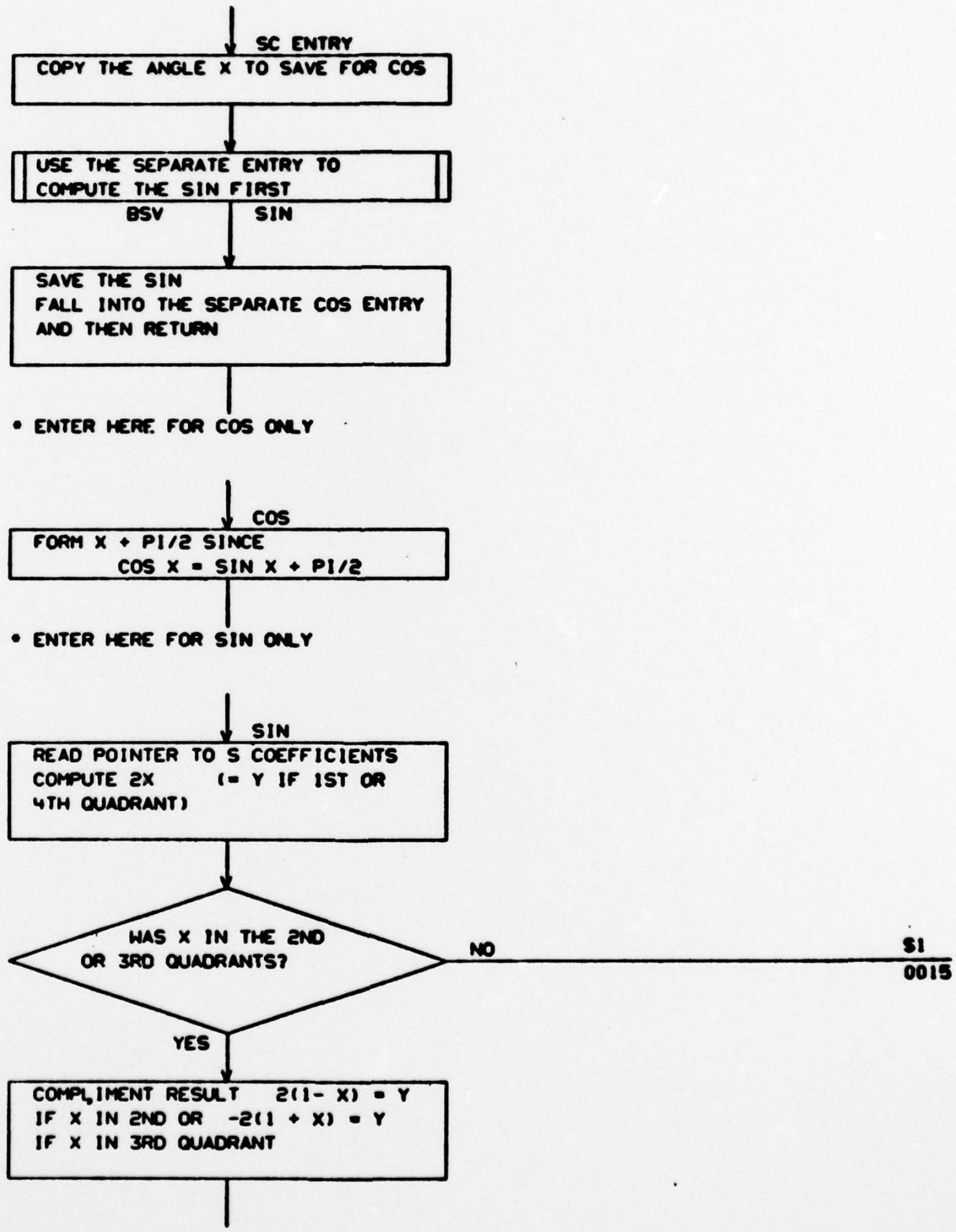
*

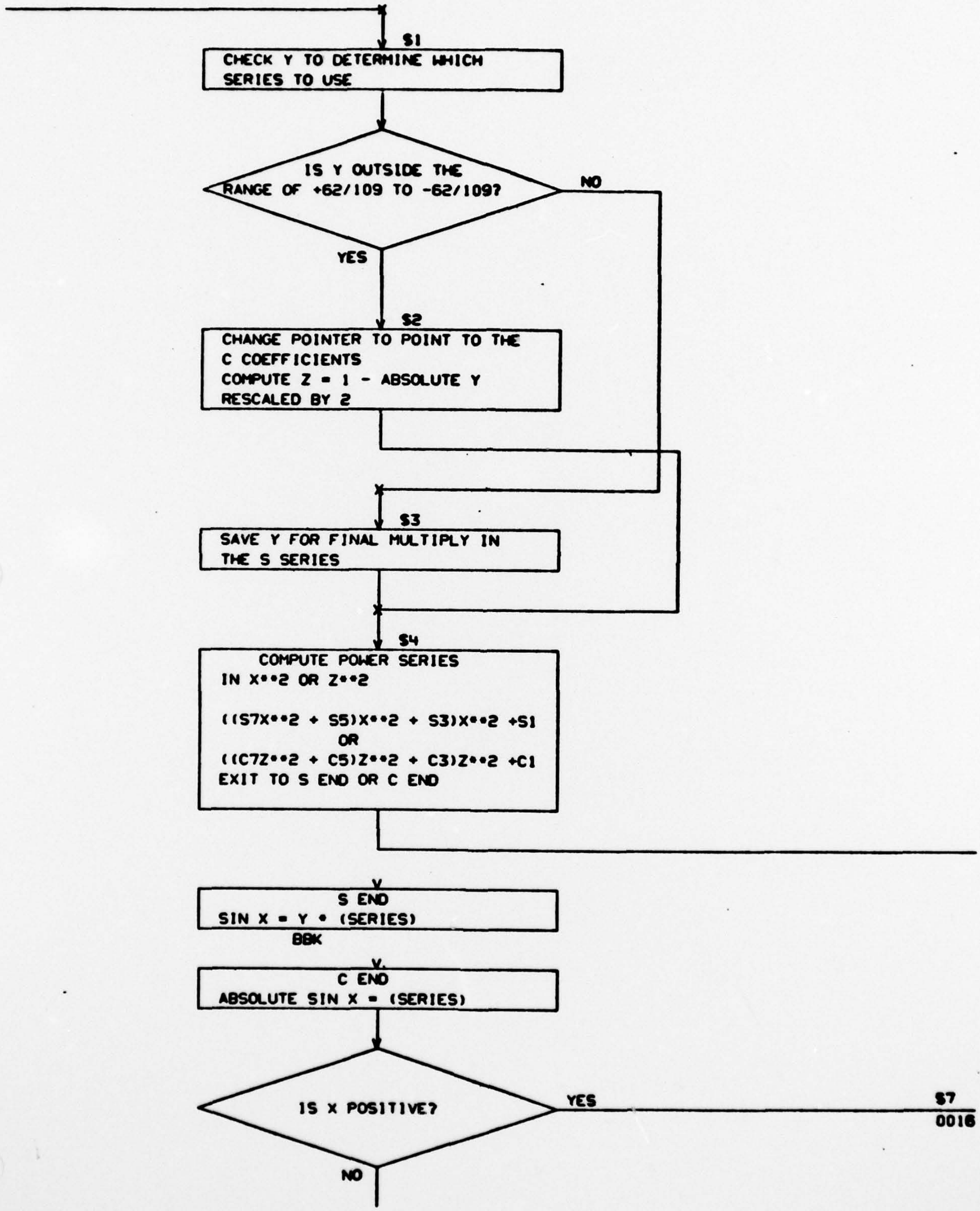
* THIS ROUTINE WILL TAKE THE SQUARE ROOT OF THE ARGUMENT. A NEGATIVE
* ARGUMENT WILL RESULT IN AN ANSWER THAT IS THE INITIAL ARGUMENT NORMA-
* LIZED. A ZERO ARGUMENT WILL GIVE A ZERO RESULT.
*




COMPUTE N/2 (TRUNCATED)
WAS N ODD?
NO
YES
MODIFY X(2) BY SQUARE ROOT OF 2
TO START NORMALIZATION FIX
S1
MULTIPLY X(2) BY 2**(-N/2)
TO GET FINAL RESULT
S2
EXIT
BBK

*   SINE - COSINE
*
* THIS SUBROUTINE WILL COMPUTE THE SIN AND COS OF ANY ANGLE SCALED PI
* THE OUTPUTS ARE SCALED B1. IT HAS SEPARATE ENTRIES FOR SIN OR COS
* ONLY.
*




S1
CHECK Y TO DETERMINE WHICH SERIES TO USE
IS Y OUTSIDE THE RANGE OF +62/109 TO -62/109?
NO
YES
S2
CHANGE POINTER TO POINT TO THE C COEFFICIENTS
COMPUTE Z = 1 - ABSOLUTE Y RESCALED BY 2
S3
SAVE Y FOR FINAL MULTIPLY IN THE S SERIES
S4
COMPUTE POWER SERIES
IN X**2 OR Z**2
((S7X**2 + S5)X**2 + S3)X**2 +S1
OR
((C7Z**2 + C5)Z**2 + C3)Z**2 +C1
EXIT TO S END OR C END
S END
SIN X = Y * (SERIES)
BBK
C END
ABSOLUTE SIN X = (SERIES)
IS X POSITIVE?
YES
S7
0016
NO

SIN X = - (SERIES)

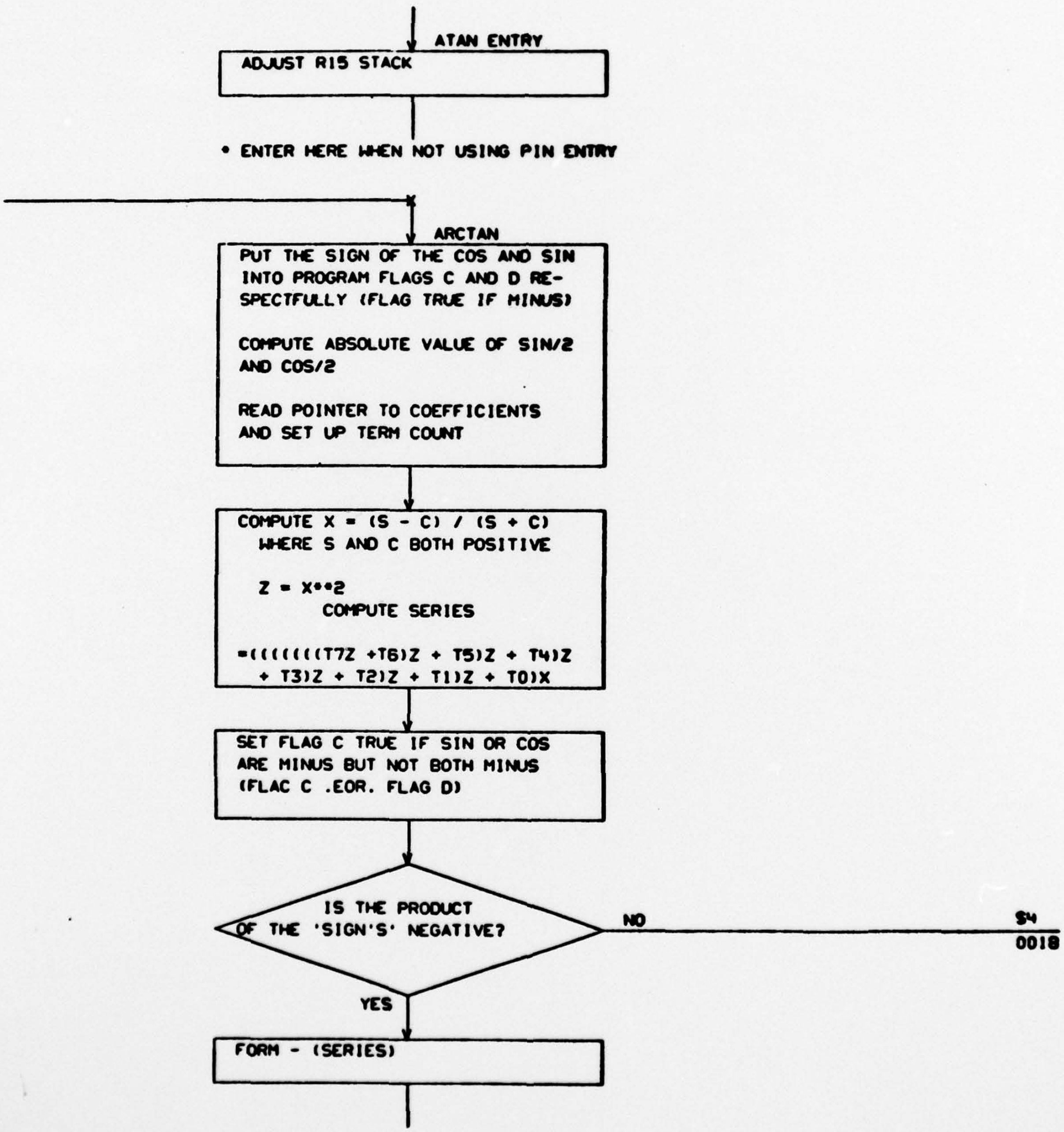
BBK

S7

SIN X = (SERIES)

BBK

*
*   ARCTANGENT
*
* THIS ROUTINE WILL COMPUTE THE ARCTANGENT FOR A SINE COSINE PAIR.
* THE SINE AND COSINE ARGUMENTS MUST HAVE THE SAME SCALE FACTOR AND THE
* RESULTING ANGLE IS SCALED PI.
*



$4

USE FLAGS D AND C +45 DEGREES TO
FIX THE QUADRANT
D TRUE = 180, C TRUE = 90 DEG

BBK

```
*                          ARCSINE
*
* THIS ROUTINE WILL COMPUTE AN ANGLE SCALED PI FROM A SINE ARGUMENT (B1)
*
```

ARCSIN

```
        COMPUTE COS
= (1 - SIN**2)**.5

SET COS = 0 IF SIN MORE THAN 1
        COMPUTE ANGLE
= ARCTAN SIN/COS

(EXIT ASIN FROM ARCTAN)
```

ARCTAN
0017

*

*   ARCCOSINE

*

* THIS ROUTINE WILL COMPUTE AN ANGLE SCALED PI FROM A COS ARGUMENT (B1)

*

ARCCOS

COMPUTE SIN
= (1 - COS**2)**-5

SET SIN = 0 IF COS MORE THAN 1

COMPUTE ANGLE
= ARCTAN SIN/COS

(EXIT ACOS FROM ARCTAN)

ARCTAN
0017

3.3 COMPUTER SUBPROGRAM ENVIRONMENT

3.3.1 Common Subroutine - Tables

a) D PHI KI Routine:

Station Burst Time Table: This table is used to determine the burst time for each station. The table is defined in detail in the flow charts (page 14 of OMEGA Processing) and in the listing. The resolution for the burst time is 0.005 second. The official name of the table is STATION BURST TIMES.

b) THETA C Routine:

Lambda Table: This table is used to convert radians to cycles at the appropriate frequency. It is defined in the listing. The official name of this table is LAMBDA TABLE.

c) THETA1 Routine:

Station Locations Table: This table contains the locations of all existing OMEGA transmitting stations. Each location is specified as a three element geocentric position vector. The first entry is for station A. The table is defined in detail in the listing. The official name of this table is STATION VECTOR TABLE.

3.3.2 Temporary Storage

All temporary storage is in the R15 push-down stack.

3.3.3 Input/Output Formats

Not applicable.

3.3.4 Required System Library Subroutines

All subroutines listed below are found in this volume.

SUBROUTINES	CALLING LABEL	FLOW DIAGRAM
ARCCOSINE	ARCCOS	p20
ARCSINE	ARCSIN	p19
ARCTANGENT	ATAN	p17 p19/1 p1/3 p20/2 p8/1
BEARING	BEARING	p8
COMPUTE THETA1	COMPUTE THETA1	p1
CROSS PRODUCT	REGISTER CROSS	p1/2 p10
DOT PRODUCT	REGISTER DOT	p1/2 p8/1 p11
D PHI KI	D PHI KI	p2/5 p3
POSITION MATRIX COMPUTATION	PANEL MAIN $8	p5
RESOLVE	RESOLVE	p7
RIJ ROTATE	ROTATE RIJ'S	p6
SIN-COS	SINCOS	p5/$8 p15
SQUARE ROOT	SQUARE ROOT	p12
THETAC	THETAC	p2
VECTOR BY MATRIX MULTIPLY	MATRIX 30	p9